



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

The Hunt for a Red Spider: Conjunctive Query Determinacy Is Undecidable

Citation for published version:

Gogacz, T & Marcinkowski, J 2015, The Hunt for a Red Spider: Conjunctive Query Determinacy Is Undecidable. in *Logic in Computer Science (LICS), 2015 30th Annual ACM/IEEE Symposium on*. Institute of Electrical and Electronics Engineers (IEEE), pp. 281 - 292. <https://doi.org/10.1109/LICS.2015.35>

Digital Object Identifier (DOI):

[10.1109/LICS.2015.35](https://doi.org/10.1109/LICS.2015.35)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

Logic in Computer Science (LICS), 2015 30th Annual ACM/IEEE Symposium on

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



The Hunt for a Red Spider: Conjunctive Query Determinacy Is Undecidable.

Tomasz Gogacz, Jerzy Marcinkowski,
Institute of Computer Science, University Of Wroclaw,

January 8th, 2015

Abstract—We solve a well known, long-standing open problem in relational databases theory, showing that the conjunctive query determinacy problem (in its "unrestricted" version) is undecidable.

I. INTRODUCTION

Imagine there is a database we have no direct access to, but there are views of this database available to us, defined by some conjunctive queries Q_1, Q_2, \dots, Q_k . And we are given another conjunctive query Q_0 . Will we be able to compute Q_0 only using the available views? The answer depends on whether the queries Q_1, Q_2, \dots, Q_k determine query Q_0 . To state it more precisely, the Conjunctive Query Determinacy Problem (CQDP) is:

The instance of the problem is a set of conjunctive queries $\mathcal{Q} = \{Q_1, \dots, Q_k\}$, and another conjunctive query Q_0 .

The question is whether \mathcal{Q} determines Q_0 , which means that for each two structures (database instances) \mathbb{D}_1 and \mathbb{D}_2 such that $Q(\mathbb{D}_1) = Q(\mathbb{D}_2)$ for each $Q \in \mathcal{Q}$, it also holds that $Q_0(\mathbb{D}_1) = Q_0(\mathbb{D}_2)$.

The technical result of this paper is:

Theorem 1: CQDP is undecidable.

It is hard to imagine a more natural problem than CQDP, and better motivated. Answering queries using views appears in various contexts, see for example [Hal01] for a survey. Or [DPT99], where the context is query evaluation plans optimization. Or – to see more recent examples – [FG12] where the view update problem is studied and [FKN13] where the context are description logics. It is fair to say that many variants of the problem are being considered, and the case we study, where both the views and the query are conjunctive queries, is not the only possible scenario. But it is of special importance as the CQs – as [NSV07] puts it – are "the simplest and most common language to define views and queries"

As we said it is hard to imagine a more natural problem than CQDP. So no wonder it has a 30 years long history

as a research subject. And this history happens to be quite complicated, marked by errors and confusion.

The oldest paper we were able to trace, where CQDP is studied, is [LY85], whose first sentence is almost the same as ours: "Assume that a set of derived relations is available in a stored form. Given a query, can it be computed from the derived relations and, if so, how?". It was shown there, and in the next paper [YL87], by the same authors, that the problem is decidable if \mathcal{Q} consists of just one query without self-joins (there is however some additional form of selection allowed there, so it is not really comparable to the CQ paradigm). Over the next 30 years many other decidable cases have been found. Let us just cite the most recent results: [NSV10] shows that the problem is decidable if each query from \mathcal{Q} has only one free variable; in [Afr11] decidability is shown for \mathcal{Q} and Q_0 being "path queries". This is generalized in [Pas11] to the scenario where \mathcal{Q} are path queries but Q_0 is any conjunctive query.

As we said in the Abstract, decidability of CQDP was a long standing open problem. It was indeed open, since 1985, but not without pauses. It was shown in [LMS95] that it is decidable whether – for given \mathcal{Q} and Q_0 like in CQDP – there exists another query Q' , over the signature consisting of Q_1, Q_2, \dots, Q_k , such that for each structure (database instance) \mathbb{D} there is $Q_0(\mathbb{D}) = Q'(\mathbb{D})$ (notice that while the "input" of Q are the relations of \mathbb{D} , which we do not have access to, the "input" of Q' are the views that we are allowed to see). Existence of such Q' – a rewriting of Q_0 – indeed implies determinacy. But – and this fact was for a long time surprisingly poorly understood – not necessarily determinacy implies existence of a rewriting. There is no sign in [LMS95] that the authors were aware of this distinction, and it seems that the first to realize that there is any problem here were the authors of [SV05]. After realizing that conjunctive query determinacy and conjunctive query rewriting (as above defined) are possibly two different notions they show that they are in fact equivalent. Together with the result of [LMS95] this would imply decidability of CQDP. But – again surprisingly – this proof was not correct, as spotted by (a superset of) the authors of [SV05] in [NSV07]. Also in [NSV07] a (correct) counterexample is presented, of \mathcal{Q} and Q_0 such that \mathcal{Q} determines Q_0 but no

rewriting Q' being a CQ exists. In fact – as it is also shown in [NSV07] – $Q_0(\mathbb{D})$ is not always a monotonic function of $Q_1(\mathbb{D}), \dots, Q_k(\mathbb{D})$.

Coming back to decidability of the determinacy problem: the paper [NSV07] is also the first to present a negative result. It was shown there, that the problem is undecidable if unions of conjunctive queries are allowed rather than CQs. In [NSV10] it was also proved that determinacy is undecidable if the elements of \mathcal{Q} are CQs and Q_0 is a first order sentence (or the other way round). Another negative result is presented in [FGZ12]: determinacy is shown there to be undecidable if \mathcal{Q} is a DATALOG program and Q_0 is CQ.

In our setting the instance of the problem consists of the set \mathcal{Q} of the queries that define the views and of the query Q_0 . A natural question to ask would be what happens if $Q_1(\mathbb{D}), \dots, Q_k(\mathbb{D})$ were also part of the input. This problem can be easily shown to be decidable. Complexity is studied in [AD98].

A. Finite vs. unrestricted case.

As usually in database theory there are two variants of the problem that one can consider: *finite*, where all the structures in question (which in our case means \mathbb{D}_1 and \mathbb{D}_2) are assumed to be finite, and *unrestricted*, where there is no such assumption. Most of the results of [LMS95], [NSV07], [NSV10], [Afr11] and [Pas11] that we report above hold true regardless of the finiteness assumption. Unlike them, Theorem 1 of this paper concerns the unrestricted case only. Decidability of CQDP for the finite case remains open.

OUTLINE

The rest of the paper is devoted to the proof of Theorem 1.

II. PRELIMINARIES

In Section II-A we recall some standard finite model theory/database theory notions. They way we present them is rather standard. In Sections II-B and II-C we also recall standard notions, but our notations may be seen as slightly non-standard (although of course equivalent to standard). This is how we think we need them in this paper.

A. Basic notions

When we say "structure" we mean a relational structure \mathbb{D} over some signature Σ , i.e. a set of elements (vertices), denoted as $Dom(\mathbb{D})$ and a set of relational atoms, whose arguments are elements of \mathbb{D} and whose predicate names are from Σ . Atoms are (of course) only positive. For an atomic formula A we use notation $\mathbb{D} \models A$ to say that A is an atom of \mathbb{D} .

Apart from predicate symbols Σ can also contain constants. If c is a constant from Σ and \mathbb{D} is a structure over Σ then $c \in Dom(\mathbb{D})$.

\mathbb{D}_1 a substructure of \mathbb{D} (and \mathbb{D} is a superstructure of \mathbb{D}_1) if for each atom A if $\mathbb{D}_1 \models A$ then $\mathbb{D} \models A$. This implies that $Dom(\mathbb{D}_1) \subseteq Dom(\mathbb{D})$.

For two structures \mathbb{D}_1 and \mathbb{D} over the same signature Σ a function $h : Dom(\mathbb{D}_1) \rightarrow Dom(\mathbb{D})$ is called a homomorphism if for each $P \in \Sigma$ of arity l and each tuple $\bar{a} \in Dom(\mathbb{D}_1)^l$ if $\mathbb{D}_1 \models P(\bar{a})$ then $\mathbb{D} \models P(h(\bar{a}))$ (where $h(\bar{a})$ is a tuple of images of elements of \bar{a}). Notice that \mathbb{D}_1 a substructure of \mathbb{D} if and only if identity is a homomorphism from \mathbb{D}_1 to \mathbb{D} .

A conjunctive query (over Σ), in short CQ, is a conjunction of atomic formulas (over Σ) whose arguments are either variables or the constants from Σ , preceded by existential quantifier binding some of the variables. It is very important in this paper to distinguish between a conjunctive query and its quantifier-free part. We usually write Ψ or Φ for a conjunction of atoms without quantifiers and Q (possibly with a subscript) for conjunctive queries, so that we have something like:

$$Q(\bar{x}) = \exists \bar{y} \Psi(\bar{y}, \bar{x})$$

where $\Psi(\bar{y}, \bar{x})$ is a formula being a conjunction of atomic formulas and \bar{x} is a tuple of variables which are free in Q .

For a conjunction of atoms Ψ (or for a CQ $Q(\bar{x}) = \exists \bar{y} \Psi(\bar{y}, \bar{x})$) the canonical structure of Ψ , denoted as $A[\Psi]$, is the structure whose elements are all the variables and constants appearing in Ψ and whose atoms are atoms of Ψ . It is useful to notice that for a structure \mathbb{D} and a set $V \subseteq Dom(\mathbb{D})$ there is a unique conjunctive query Q such that $\mathbb{D} = A[Q]$ and that V is the set of free variables of Q .

For a CQ $Q(\bar{x}) = \exists \bar{y} \Psi(\bar{y}, \bar{x})$ with $\bar{x} = x_1, \dots, x_l$, a structure \mathbb{D} and a tuple a_1, \dots, a_l of elements of \mathbb{D} we write $\mathbb{D} \models Q(a_1, \dots, a_l)$ when there exists a homomorphism $h : A[\Psi] \rightarrow \mathbb{D}$ such that $h(x_i) = a_i$ for each i .

Sometimes we also write $\mathbb{D} \models Q$. Then we assume that all the free variables of Q are implicitly existentially quantified, so that the meaning of the notation is that there exists any homomorphism $h : A[\Psi] \rightarrow \mathbb{D}$.

The most fundamental definition of this paper now, needed to formulate the problem we solve: for a CQ Q and for a structure \mathbb{D} by $Q(\mathbb{D})$ we denote the "view defined by Q over \mathbb{D} ", which is the relation $\{\bar{a} : \mathbb{D} \models Q(\bar{a})\}$.

B. TGDs and how they act on a structure

A Tuple Generating Dependency (or TGD) is a formula of the form:

$$\forall \bar{x}, \bar{y} \Phi(\bar{x}, \bar{y}) \Rightarrow \exists \bar{z} \Psi(\bar{z}, \bar{y})$$

where Ψ and Φ are – as always – conjunctions of atomic formulas. The standard convention, which we will obey, is that the universal quantifiers in front of the TGD are omitted.

From the point of view of this paper it is important to see a TGD – let it be T , equal to $\Phi(\bar{x}, \bar{y}) \Rightarrow \exists \bar{z} \Psi(\bar{z}, \bar{y})$ – as a procedure whose input is a structure \mathcal{D} and whose output is a new structure being a superstructure of \mathcal{D} :

find a tuple \bar{b} (with $|\bar{b}| = |\bar{y}|$) such that:

- ① $\mathcal{D} \models \exists \bar{x} \Phi(\bar{x}, \bar{b})$ via homomorphism h but
- ② $\mathcal{D} \not\models \exists \bar{z} \Psi(\bar{z}, \bar{b})$;

create a new copy of $A[\Psi]$;

output $T(\mathcal{D}, \bar{b})$ being a union of \mathcal{D} and the new copy of $A[\Psi]$, with each y from $A[\Psi]$ identified with $h(y)$ in \mathcal{D} .

The message, which will be **good to remember**, is that the interface between the “new” part of the structure, added by a single application of a TGD to a structure, and the “old” structure, are the free variables of the query in the right hand side of the TGD.

C. Chase and its universality

For a structure \mathcal{D} and a set \mathcal{T} of TGDs let $\text{ToDo}(\mathcal{T}, \mathcal{D})$ be the set of all pairs $\langle \bar{b}, T \rangle$ such that T , equal to $\Phi(\bar{x}, \bar{y}) \Rightarrow \exists \bar{z} \Psi(\bar{z}, \bar{y})$, is a TGD from \mathcal{T} and \bar{b} satisfies conditions ① and ②. Roughly speaking $\text{ToDo}(\mathcal{T}, \mathcal{D})$ is the set of tuples of elements of \mathcal{D} which satisfy the left hand side of some TGD in \mathcal{T} but still wait for a witness – confirming that they also satisfy the right hand side – to be added to the structure.

A sequence $\{\mathcal{D}_i\}_{i \in \Omega}$ of structures, for some ordinal number Ω , will be called *fair* (with respect to \mathcal{T} and \mathbb{D}) if:

- $\mathcal{D}_0 = \mathbb{D}$;
- for each $i > 0$ we have $\mathcal{D}_i = T(\bigcup_{j < i} \mathcal{D}_j, \bar{b})$ for some $\langle \bar{b}, T \rangle \in \text{ToDo}(\mathcal{T}, \bigcup_{j < i} \mathcal{D}_j)$;
- for each $\langle \bar{b}, T \rangle \in \text{ToDo}(\mathcal{T}, \bigcup_{j < i} \mathcal{D}_j)$ for some i , there is $k > i$ such that $\langle \bar{b}, T \rangle \notin \text{ToDo}(\mathcal{T}, \bigcup_{j < k} \mathcal{D}_j)$.

Let $\{\mathcal{D}_i\}_{i \in \Omega}$ be a fair sequence (with respect to \mathcal{T} and \mathbb{D}). Then the structure $\text{Chase}(\mathcal{T}, \mathbb{D})$ is defined as $\bigcup_{i \in \Omega} \mathcal{D}_i$ and each of the sets \mathcal{D}_i is called a *stage of Chase*.

In other words, $\text{Chase}(\mathcal{T}, \mathbb{D})$ is a structure being result of adding, one by one, tuples that witness that some TGD from \mathcal{T} is satisfied for a given tuple from the current structure. The set ToDo always contains tuples that do not have the required witnesses yet. Notice that there are two possible ways, for a tuple $\langle \bar{b}, T \rangle$, to disappear from the set ToDo : one is that the TGD T is applied to the tuple \bar{b} at some step. But it may also happen that the witnesses \bar{b} needs are added as a side-effect of other TGDs being applied to other tuples¹.

It may appear, and not without a reason, that the structure $\text{Chase}(\mathcal{T}, \mathbb{D})$ depends on the ordering in which tuples are selected from the ToDo set. But the beautiful fact (and a well-known one, [JK82]) is that, regardless of the ordering:

¹A reader who is aware of the difference between standard and oblivious Chase will notice that what we define is the standard/lazy version.

Theorem 2 (Chase as universal structure):

- $\text{Chase}(\mathcal{T}, \mathbb{D}) \models \mathcal{T}$. In other words if $\Phi(\bar{x}, \bar{y}) \Rightarrow \exists \bar{z} \Psi(\bar{z}, \bar{y})$ is a TGD from \mathcal{T} and $\text{Chase}(\mathcal{T}, \mathbb{D}) \models \exists \bar{x} \Phi(\bar{x}, \bar{b})$ then also $\text{Chase}(\mathcal{T}, \mathbb{D}) \models \exists \bar{z} \Psi(\bar{z}, \bar{b})$
- Let \mathbb{M} be any superstructure of \mathbb{D} such that $\mathbb{M} \models \mathcal{T}$ and let Q be any conjunctive query such that $\text{Chase}(\mathcal{T}, \mathbb{D}) \models Q$. Then also $\mathbb{M} \models Q$.

Most of the lemmas in this paper, concerning the structure of $\text{Chase}(\mathcal{T}, \mathbb{D})$ for specific \mathcal{T} and \mathbb{D} are proved by induction on a respective fair sequence, even if this is not always mentioned explicitly.

D. Thue systems

Our undecidability proof is by reduction from a variant of the Thue systems word problem (also known as semigroups word problem). A Thue system is given by a finite symmetric relation $\Pi \subseteq \mathcal{A}^* \times \mathcal{A}^*$ for some finite alphabet \mathcal{A} . For two words $\mathbf{w}, \mathbf{w}' \in \mathcal{A}^*$ we define $\mathbf{w} \Leftrightarrow_{\Pi} \mathbf{w}'$ if and only if there are words $\mathbf{v}, \mathbf{v}' \in \mathcal{A}^*$ and a pair $\{\mathbf{t}, \mathbf{t}'\} \in \Pi$ such that $\mathbf{w} = \mathbf{v}\mathbf{t}\mathbf{v}'$ and $\mathbf{w}' = \mathbf{v}\mathbf{t}'\mathbf{v}'$. Relation $\stackrel{*}{\Leftrightarrow}_{\Pi}$ is defined as the transitive closure of \Leftrightarrow_{Π} . Various undecidability results involving relation $\stackrel{*}{\Leftrightarrow}_{\Pi}$ can be proved using standard techniques from [Dav77].

III. GREEN-RED TGDs

A. Green-Red Signature

For a given signature Σ let Σ_G and Σ_R be two copies of Σ having new relation symbols, which have the same names and the same arities as symbols in Σ but are written in green and red respectively. Let $\bar{\Sigma}$ be the union of Σ_G and Σ_R . Notice that the constants from Σ , not being relation symbols, are never colored and thus survive in $\bar{\Sigma}$ unharmed.

For any formula Ψ over Σ let $R(\Psi)$ (or $G(\Psi)$) be the result of painting all the predicates in Ψ red (green). For any formula Ψ over $\bar{\Sigma}$ let $\text{dalt}(\Psi)$ (“daltonisation of Ψ ”) be a formula over Σ being the result of erasing the colors from predicates of Ψ . The same convention applies to structures. Whenever an uncolored relation symbol (usually H) is used in the context of $\bar{\Sigma}$ it should be understood as “ $G(H)$ ” or “ $R(H)$ ”.

B. Having \mathbb{D} instead of \mathbb{D}_1 and \mathbb{D}_2 .

We prefer to restate CQDP a little bit in order to be talking about one database instance instead of two. Clearly CQDP is equivalent to:

The green-red conjunctive query determinacy problem (GRCQDP). The instance of the problem is a finite set \mathcal{Q} of conjunctive queries and another conjunctive query Q_0 , all of them over some signature Σ . The question is whether for each structure \mathbb{D} over $\bar{\Sigma}$ such that:

- ③ $(G(Q))(\mathbb{D}) = (R(Q))(\mathbb{D})$ for each $Q \in \mathcal{Q}$
it also holds that $(G(Q_0))(\mathbb{D}) = (R(Q_0))(\mathbb{D})$.

For a conjunctive query Q of the form $\exists \bar{x} \Phi(\bar{x}, \bar{y})$ where Φ is a conjunction of atoms over Σ let $Q^{G \rightarrow R}$ be the TGD generated by Q in the following sense:

$$Q^{G \rightarrow R} = \forall \bar{x}, \bar{y} [G(\Phi)(\bar{x}, \bar{y}) \Rightarrow \exists \bar{z} R(\Phi)(\bar{z}, \bar{y})]$$

TGD $Q^{R \rightarrow G}$ is defined in an analogous way. For a set \mathcal{Q} as above let $\mathcal{T}_{\mathcal{Q}}$ be the set of all TGDs of the form $Q^{G \rightarrow R}$ or $Q^{R \rightarrow G}$ with $Q \in \mathcal{Q}$. It is very easy to see that:

Lemma 3: The above condition ③ is satisfied by structure \mathbb{D} if and only if $\mathbb{D} \models \mathcal{T}_{\mathcal{Q}}$.

Now GRCQDP can be again restated as:

Given a set \mathcal{Q} (as in the original formulation of GRCQDP, above), and another conjunctive query Q_0 , is it true that:

- ④ for each structure \mathbb{D} and each tuple \bar{a} of elements of \mathbb{D} , if $\mathbb{D} \models \mathcal{T}_{\mathcal{Q}}, G(Q_0)(\bar{a})$ then also $\mathbb{D} \models R(Q_0)(\bar{a})$?

But ④ means that $\mathcal{T}_{\mathcal{Q}}, G(Q_0)(\bar{a}) \models R(Q_0)(\bar{a})$ where \bar{a} is a tuple of new constants. Thus – by Theorem 2 – CQDP is equivalent to²

CQDP – the green-red Chase version (CQDP-GRC). Given the set \mathcal{Q} (as in the original formulation, above), and another conjunctive query Q_0 , is it true that:

$$\text{Chase}(\mathcal{T}_{\mathcal{Q}}, A[G(Q_0)(\bar{a})]) \models R(Q_0)(\bar{a}) \quad ?$$

where $A[G(Q_0)(\bar{a})]$ is the canonical structure of $G(Q_0)(\bar{a})$.

The **main technical result of this paper** is:

Theorem 4 (Theorem 1 restated): CQDP-GRC is undecidable.

Of course the problem to determine, for given set \mathcal{T} of TGDs, database instance \mathbb{D} and query Q , whether $\text{Chase}(\mathcal{T}, \mathbb{D}) \models Q$, is undecidable in general. But this does not *a priori* mean that CQDP-GRC is undecidable, since the TGDs we allow here are of very special green-red form

² The observation that determinacy can be semi-decided using chase is not ours and is at least as old as [NSV07]. The difference is that in [NSV07] they prefer to see two separate structures rather than two colors.

(with the head being just recoloring of the body) and since we only consider Q being a recoloring of \mathbb{D} .

C. Idempotence lemma

One useful feature of the green-red TGDs is described in the following easy lemma:

Lemma 5: Let \mathcal{Q} be a set of conjunctive queries and let the set $\mathcal{T}_{\mathcal{Q}}$ of the green-red TGDs generated by \mathcal{T} be defined as before. Let T be $Q^{R \rightarrow G}$ for some $Q \in \mathcal{Q}$ and suppose $\bar{b} \in \mathcal{D}$ is such that $\langle \bar{b}, T \rangle \in \text{ToDo}(\mathcal{T}_{\mathcal{Q}}, \mathcal{D})$. Suppose $\mathcal{D}' = T(\mathcal{T}, \bar{b})$. Then $\langle \bar{b}, Q^{G \rightarrow R} \rangle \notin \text{ToDo}(\mathcal{T}_{\mathcal{Q}}, \mathcal{D}')$

Proof: The necessary condition for $\langle \bar{b}, Q^{R \rightarrow G} \rangle$ to be in $\text{ToDo}(\mathcal{T}_{\mathcal{Q}}, \mathcal{D})$ is that $\mathcal{D} \models R(Q)(\bar{b})$. Since \mathcal{D}' is a superstructure of \mathcal{D} we also have $\mathcal{D}' \models R(Q)(\bar{b})$. But the necessary condition for $\langle \bar{b}, Q^{G \rightarrow R} \rangle$ to be in $\text{ToDo}(\mathcal{T}_{\mathcal{Q}}, \mathcal{D})$ is that $\mathcal{D}' \not\models R(Q)(\bar{b})$. \square

Of course both the lemma and its proof also hold for the colors reversed.

OUTLINE

The rest of this paper is devoted to the proof of Theorem 4. The proof is by encoding the word problem for some very specific Thue systems over a very specific alphabet (being a subset of) \mathbb{A}_s .

In Section IV we study s-piders, which are elements of the set \mathbb{A}_s , and s-pider queries \mathbb{F}_s which are partial functions from \mathbb{A}_s to \mathbb{A}_s .

Then, in Section V and later, we show how to concatenate s-piders into words, and how to modify \mathbb{F}_s to get functions that take a pair of elements of \mathbb{A}_s as an input, and output pairs of elements of \mathbb{A}_s . This opens the way to Thue systems encoding.

IV. S-PIDERS AND GRAPH REACHABILITY

Let $s \in \mathbb{N}$ be fixed and let Σ be a signature consisting of:

- constants c_1, c_2, \dots, c_s and c^1, c^2, \dots, c^s
- binary relation symbols C_1, C_2, \dots, C_s and C^1, C^2, \dots, C^s (the C reads as "calf" here)
- binary relation symbols T_1, T_2, \dots, T_s and T^1, T^2, \dots, T^s (the T reads as "thigh")
- ternary relation symbol H .

Σ_G, Σ_R and $\bar{\Sigma}$ are defined as in Section III.

For an element a of a structure \mathbb{D} over $\bar{\Sigma}$ by out-degree of a we mean the number of atoms $P(a, b)$, with $P \in \bar{\Sigma}$ and $b \in \mathbb{D}$, which are true in \mathbb{D} . The in-degree is defined in an analogous way. By out-degree of a with respect to P , with $P \in \Sigma \cup \bar{\Sigma}$ we mean the number of atoms $P(a, b)$, with $b \in \mathbb{D}$, which are true in \mathbb{D} .

From now on i, j are **always** natural numbers from the set $\mathbb{S} = \{1, 2, \dots, s\}$. Another notation we use is $I, J \subseteq \mathbb{S}$ which **always** mean either a singleton or the empty set.

Being computer scientists, we do not distinguish between a singleton and its only element.

A. S-piders and their taxonomy

For a conjunction of atomic formulas Ψ and for an atom P (atoms P, P') occurring in Ψ let Ψ/P (resp. $\Psi/P, P'$) be Ψ with P (resp. P and P') removed from the conjunction.

Let now Φ_s be defined as the following conjunction of atomic formulas:

$$H(z, z_1, z_2) \wedge \bigwedge_{i=1}^s T_i(z, x_i) \wedge T^i(z, y_i) \wedge C_i(x_i, c_i) \wedge C_i(y_i, c^i)$$

Definition 6: • The ideal green full s-pider, denoted as \star , is $A[G(\Phi_s)]$ – the canonical structure of the green version of Φ_s . The ideal red full s-pider r , denoted as \star^i , is $A[R(\Phi_s)]$.

• An ideal green 1-lame upper s-pider, denoted \star^i , is $A[G(\Phi_s/C^i(y_i, c^i)) \wedge R(C^i(y_i, c^i))]$. An ideal red 1-lame upper s-pider, denoted \star^i , is $A[R(\Phi_s/C^i(y_i, c^i)) \wedge G(C^i(y_i, c^i))]$.

• An ideal green 1-lame lower s-pider, denoted \star_i , is $A[G(\Phi_s/C_i(x_i, c_i)) \wedge R(C_i(x_i, c_i))]$. An ideal red 1-lame lower s-pider, denoted \star_i , is $A[R(\Phi_s/C_i(x_i, c_i)) \wedge G(C_i(x_i, c_i))]$.

• An ideal green 2-lame s-pider, denoted \star_j^i , is $A[G(\Phi_s/C_i(x_i, c_i), C^j(y_j, c^j)) \wedge R(C_i(x_i, c_i) \wedge C^j(y_j, c^j))]$.

• An ideal red 2-lame s-pider, denoted \star_j^i , is $A[R(\Phi_s/C_i(x_i, c_i), C^j(y_j, c^j)) \wedge G(C_i(x_i, c_i) \wedge C^j(y_j, c^j))]$.

Notice that each of ideal s-piders really looks exactly like a spider: there is a *head* (z), with $2s$ legs attached to it; each leg has length 2, and the legs are distinguishable. Head is connected to the *tail* (z_1) and the *antenna* (z_2). But the antenna and the tail will not bother us in this Section.

Full s-piders – red and green – are monochromatic, head and all legs must be of the same color. 1-lame s-piders have one calf of the opposite color. As we distinguish between the “upper” and “lower” legs of a s-pider, we have two kinds of 1-lame s-piders of each color. A 2-lame s-pider has one upper calf and one lower calf of the opposite color. Any of the $2s$ vertices of a s-pider which are neither head nor a constant will be called a *knee*. Sometimes we will need to be more precise, and talking about particular s-pider we will use descriptions like “ i ’th upper knee”, hoping that meaning of it is clear.

Definition 7: \mathbb{A}_s is the set of all \star_j^I and \star_j^I , with I and J as defined above. A s-pider \star_j^I (or \star_j^I) is called upper if I is non-empty and is called lower if J is non-empty.

In other words \mathbb{A}_s is the set of all ideal s-piders: full, 1-lame and 2-lame, both red and green. Notice that a 1-lame s-pider is always either upper or lower, a 2-lame s-pider is both, and a full s-pider is neither upper nor lower.

While ideal s-piders are finitely many ($2 + 4s + 2s^2$ of them), for each structure over $\bar{\Sigma}$ there can be many – maybe

infinitely many – actual incarnations of ideal s-piders in this structure:

Definition 8: A real s-pider is any structure \mathcal{S} (in particular a substructure of another structure) such that:

- $\text{dalt}(\mathcal{S}) \models \Phi_s$,
- if \mathcal{S}' is a proper substructure of \mathcal{S} then $\text{dalt}(\mathcal{S}') \not\models \Phi_s$.

The second condition of the above definition looks more complicated than it really is. We just do not want a house full of s-piders to be called a s-pider.

B. S-pider queries and what they are good for

Let us first remind the reader that for each structure \mathbb{D} , and each subset V of $\text{Dom}(\mathbb{D})$ there exists a unique conjunctive query Ψ such that V is the set of free variables of Ψ and $\mathbb{D} = A[\Psi]$:

Definition 9 (s-pider queries): 1) \mathbb{f}_j^i is the unique query with free variables x_j and y_i whose canonical structure is equal to $A[\Phi_s/C_j(x_j, c_j), C^i(y_i, c^i)]$;

2) \mathbb{f}^i is the unique query with single free variable y_i whose canonical structure is equal to $A[\Phi_s/C^i(y_i, c^i)]$

3) \mathbb{f}_i is the unique query with single free variable x_i whose canonical structure is equal to $A[\Phi_s/C_i(x_i, c_i)]$

By analogy with s-piders, the s-pider queries of the form \mathbb{f}_j^i will be sometimes called 2-lame, and of the form \mathbb{f}^i or \mathbb{f}_j will be called 1-lame. And, like 1-lame s-piders, also 1-lame s-pider queries are either upper and lower, while 2-lame are both. Let \mathbb{F}_s be the set of all s-pider queries. Let us now learn – by examples – how the green-red TGDs generated by the queries from \mathbb{F}_s act on elements of \mathbb{A}_s .

Example 1. Suppose \mathcal{Q} consist of a single query \mathbb{f}_j^i for some i, j and let $\mathcal{T}_{\mathcal{Q}}$ be the set of TGDs, as defined in Section III. Let us try to understand how the TGDs of $\mathcal{T}_{\mathcal{Q}}$ can be applied to \star^i .

$\mathcal{T}_{\mathcal{Q}}$ consists of two TGDs. One of them is $(\mathbb{f}_j^i)^{R \rightarrow G}$.

It tries to find, in the current structure, a homomorphic image \mathcal{D} of $A[R(\mathbb{f}_j^i)]$ and, if this succeeds, it:

- produces a fresh copy of $A[G(\mathbb{f}_j^i)]$ and
- identifies elements of this copy resulting from free variables of $G(\mathbb{f}_j^i)$ with elements of \mathcal{D} resulting from the respective free variables³ in $R(\mathbb{f}_j^i)$.

The other TGD, $(\mathbb{f}_j^i)^{G \rightarrow R}$, does the same, but with the colors reversed.

Now, if the current structure is \star^i , which is red, then of course the only possible match is with $(\mathbb{f}_j^i)^{R \rightarrow G}$. The s-pider \star^i is lame, it lacks his upper i -th calf⁴ but it is not needed for a match since $G(\mathbb{f}_j^i)$ lacks this calf too.

³Of course the constants from the language are seen as free variables here, and their different occurrences are also identified.

⁴Actually it has one, but green, and the atoms in the body of any TGD of the form $Q^{R \rightarrow G}$ are red, so they only can match with red atoms.

Thus a new – green – copy $G(A[\mathbb{f}_j^i])$ of $A[\mathbb{f}_j^i]$ will be created. How will it be connected to the original \star^i ?

Of course all the constants from $G(A[\mathbb{f}_j^i])$ (which means all constants from Σ apart from c^i and c_j) will be identified with the respective constants in \star^i . Also the i -th upper knee of $G(A[\mathbb{f}_j^i])$ will be identified with the respective knee of \star^i and the j -th lower knee of $G(A[\mathbb{f}_j^i])$ will be identified with the respective knee of \star^i . Notice that, while $G(A[\mathbb{f}_j^i])$ is not a s-pider (it is two calves short of being one), we actually created a new s-pider. It consists of the copy of $G(A[\mathbb{f}_j^i])$, and of two calves that it shares with \star^i : the i -th upper calf of \star^i (which is green) and of the lower j -th calf of \star^i (which is red, and is the only red calf of the new s-pider). Not only we created a new s-pider but we already have a name for it – it is a copy of \star_j^i ! We cannot resist the temptation of writing this as:

$$\mathbb{f}_j^i(\star^i) = \star_j^i$$

Example 2. Let now \mathcal{Q} consist of a single query \mathbb{f}_j^i and consider a s-pider \star_k with $k \neq j$. Since \star_k is green, there is of course no match with the left hand side of the TGD $(\mathbb{f}_j^i)^{R \rightarrow G}$. But is there a match with the left hand side of $(\mathbb{f}_j^i)^{G \rightarrow R}$? Notice that the atom $G(C_k(x_k, c_k))$ occurs in $G(\mathbb{f}_j^i)$. But not in \star_k – the k 'th lower calf of \star_k is red. We cannot resist the temptation of writing this as:

$$\text{if } k \neq j \text{ then } \star_k \notin \text{Dom}(\mathbb{f}_j^i)$$

Example 3. Let again \mathcal{Q} be the single query \mathbb{f}_j^i for some i, j . We already know that $\mathbb{f}_j^i(\star^i) = \star_j^i$ or, to be more precise, that $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star^i) \models \star_j^i$. By exactly the same argument we get that:

- $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star_j^i) \models \star^i$,
- $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star^i) \models \star_j^i$,
- $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star_j^i) \models \star^i$.

Definition 10: Suppose $\mathcal{S}, \mathcal{S}' \in \mathbb{A}_s$ are such that $\mathcal{S} \neq \mathcal{S}'$. Let $X \in \{G, R\}$ be the of color of \mathcal{S} and let Y be the opposite color. Let T be the TGD: $X(f)(\bar{w}, \bar{u}) \Rightarrow \exists \bar{v} Y(f)(\bar{v}, \bar{u})$.

Then by $f(\mathcal{S}) = \mathcal{S}'$ we will mean that:

- $\mathcal{S} \models (X(f))(\bar{b})$ for some tuple \bar{b} of elements of \mathcal{S} ;
- \mathcal{S}' is a substructure of $T(\mathcal{S}, \bar{b})$

In other words, $f(\mathcal{S}) = \mathcal{S}'$ means that one of the two green-red TGDs generated by f can be applied to \mathcal{S} and that a copy of \mathcal{S}' is then produced in one step. It is of course easy to see that the color of \mathcal{S}' is then Y .

The examples of the previous subsection can be easily extended to a proof of:

Lemma 11 (Algebra of s-piders): Let $I, J, I', J' \subseteq \mathbb{S}$ be as before. Then $\mathbb{f}_J^I(\star_{J'}^{I'})$ is defined if and only if $I' \subseteq I$ and $J' \subseteq J$. If this the case then $\mathbb{f}_J^I(\star_{J'}^{I'}) = \star_{J \setminus J'}^{I \setminus I'}$.

The same holds for the colors reversed.

C. Example: Encoding graph reachability

As one more toy example, consider an undirected graph $\langle V, E \rangle$, with $V = \{v_1, v_2, \dots, v_t\}$ and $E = \{e_1, e_2, \dots, e_t\}$. Suppose – for simplicity of presentation – that degree of v_1 is exactly 1, and that e_1 is the only edge containing v_1 .

Let $s \in \mathbb{N}$ be such that $s \geq t$ and $s \geq t'$ and let the set \mathcal{Q} contain the following s-pider queries: $\mathbb{f}_1, \mathbb{f}^2$ and, for each triple i, j, k such that $e_k = \{v_i, v_j\}$, two queries: \mathbb{f}_k^i and \mathbb{f}_k^j .

Now we can represent graph reachability as an instance of GRCQDP:

Observation 12: The two conditions are equivalent:

- (i) There is a path, in $\langle V, E \rangle$, from v_1 to v_2 ;
- (ii) $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star) \models \star$

For the proof⁵, suppose that there is a path $v_1, e_1, v_{i_1}, e_{i_1} \dots v_{i_t}, e_{i_t}, v_2$ from v_1 to v_2 . One can see that $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star) \models \star$ contains the following s-piders: \star_1 (produced from \star by \mathbb{f}_1), $\star_1^{i_1}$ (produced from \star_1 by $\mathbb{f}_1^{i_1}$), \star_{i_1} , and so on. For each vertex v_k reachable from v_1 the green 1-lame upper s-pider \star_k^i will be at some point added to the chase and for each edge e_k reachable from v_1 the red 1-lame lower s-pider \star_k^j will be added. Finally, once we have \star^2 , the query \mathbb{f}^2 can be used to produce \star .

But wait: how about the opposite direction? Clearly, the queries of \mathcal{Q} were designed to only produce the red s-piders for reachable edges and green for reachable vertices (as above) but how are we sure that there are no side-effects leading to the creation of \star even if v_2 is not reachable from v_1 ? There could be two sources of such side-effects. One is that – due to the complicated structure of $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$ new real s-piders could emerge there⁶, which were not produced as $f(\mathcal{S})$, for $f \in \mathcal{Q}$ and \mathcal{S} previously in $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$. This could in principle happen, the s-piders share constants, and knees and who knows what more.

Second possible source of problems is that some weird application of queries from \mathcal{Q} to the s-piders we ourselves produced could lead to creation of something more than just the representations of reachable vertices and edges (as described above).

As it turns out – and as we are going to show before the end of this Section – there are no side-effects of the first sort and while there indeed are some side-effects of the second sort, but they are "sterile" and thus controllable.

D. Understanding the structure of $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$

We want to make sure that our abstraction of low-level structures, like s-piders and TGDs, as high-level objects, as symbols \mathbb{A}_s and partial functions $f_J^I : \mathbb{A}_s \rightarrow \mathbb{A}_s$ is correct, in the sense that there are no uncontrollable side-effects. And this is what the following series of lemmas is about.

⁵Remember, this is an example, so the goal is to see the mechanisms rather than a rigorous proof.

⁶To be more precise, what we really fear here are not new s-piders but new – unintended – matchings with left hand side of some TGD from $\mathcal{T}_{\mathcal{Q}}$. See Lemma 14 (iv).

Let $\mathcal{Q} \subseteq \mathbb{F}_s$. We are going to analyze the structure of $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$. Let $\{\mathcal{D}_k\}_{k \in \Omega}$ be a fair sequence (with respect to $\mathcal{T}_{\mathcal{Q}}$ and \star) and recall (see Section II-C) that $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$ is defined as $\bigcup_{k \in \Omega} \mathcal{D}_k$. Our basic proof technique will be induction on k .

Lemma 13: Each knee in $\text{Chase}(\mathcal{F}_{\mathcal{Q}}, \star)$ has out-degree 1. Each red head has out-degree 1 with respect to any red T_i and to any red T^i and out-degree 0 with respect to any other relation. The same is true, with colors reversed, for green heads.

Proof: Induction. For the induction step notice that atoms of relations C_i , C^i , T_i , T^i can only be created by the TGDs from $\mathcal{T}_{\mathcal{Q}}$ together with their leftmost argument. This means that an application of a rule from $\mathcal{T}_{\mathcal{T}}$ can never add an outgoing edge to an already existing element (notice that this is not true about in-coming edges, and this why s-piders can share a calf). \square

Lemma 14 (No low-level side-effects): Suppose an element a in $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$ is such that $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star) \models H(a, a_1, a_2)$, for some a_1, a_2 . Then:

- (i) *There exists exactly one real s-pider S in $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$ such that a is the head of S .*
- (ii) *S is created together with a , which means that if $\mathcal{D} \in \{\mathcal{D}_k\}_{k \in \Omega}$ is such that $a \in \text{Dom}(\mathcal{D})$ then $\mathcal{D} \models S$.*
- (iii) *There is an $S' \in \mathbb{A}_s$ such that S and S' are isomorphic.*
- (iv) *Suppose $f \in \mathbb{F}_s$ and h is a homomorphism from $A[R(f)]$ (or $A[G(f)]$) to $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$ such that $h(z) = a$. Then there exists a homomorphism from $A[R(f)]$ (resp. $A[G(f)]$) to S' .*

The sense of Claim (iii) is that *a priori* S could have more than just two calves of the color that is opposite to the color of its head, and then, still being a real s-pider, it would not be isomorphic to anything in \mathbb{A}_s . Proof of the Lemma (which we skip) is straightforward induction, using Lemma 13.

Lemma 15: Let $\text{zoo}(\mathcal{Q})$ be the set of all s-piders $S \in \mathbb{A}_s$ which are isomorphic to some real s-pider in $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$. Then $\text{zoo}(\mathcal{Q})$ is the smallest subset of \mathbb{A}_s containing \star and closed under functions from \mathcal{Q} .

Proof: We know, from Lemma 11 that if $S' = f(S)$ for some $f \in \mathcal{Q}$ and some $S \in \mathbb{A}_s$ then $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star) \models S'$. To see that $\text{zoo}(\mathcal{Q})$ is closed under functions from \mathcal{Q} notice that if $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star) \models S$ then $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, S)$ is a substructure of $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$ and, in consequence, also $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star) \models S'$. For the opposite direction use Lemma 14. \square

One more lemma we will need is:

Lemma 16: Suppose each query in \mathcal{Q} is lower. Then a s-pider in $\text{zoo}(\mathcal{Q})$ is red if and only if it is lower.

Proof: By usual induction on the fair sequence.

E. Idempotence and sterile s-piders

Notice that it very well may be the case that more than one copy of some $S \in \mathbb{A}_s$ will be created in $\text{Chase}(\mathcal{T}_{\mathcal{F}}, \star)$.

Let for example \mathcal{Q} be $\{\mathbb{f}_1, \mathbb{f}_2, \mathbb{f}_1^3, \mathbb{f}_2^3\}$. Then a copy of \star^3 can be constructed by first applying \mathbb{f}_1 to \star , and then \mathbb{f}_1^3 to the resulting \star_1 . But a different copy of \star^3 will be produced by first applying \mathbb{f}_2 to \star , and then \mathbb{f}_2^3 to the resulting \star_2 .

Imagine however that, after constructing (a copy S of) \star_3 in the first way, as $\mathbb{f}_1^3(\mathbb{f}_1(\star))$ we try to apply \mathbb{f}_1^3 to S . According to Lemma 11 it is of course possible, and the result is a copy of \star_1 . But it is not a new copy: it follows from Lemma 5 that second consecutive use of TGDs generated by the same query does not add to the Chase. In this context the following lemma will be particularly useful:

Lemma 17 (2-lame s-piders are sterile): Suppose S is a real 2-lame s-pider in some stage \mathcal{D} of $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$. Then S will never be used as a left hand side of a TGD execution leading to one of the later stages.

Proof: Suppose S is isomorphic to \star_j^i (the proof does not change if S is green). It follows from Lemma 11 that S could only be a result of applying the TGD $(\mathbb{f}_j^i)^{G \rightarrow R}$ to \star . But the only TGD that matches with S is $(\mathbb{f}_j^i)^{R \rightarrow G}$. By Lemma 5 it cannot however be now applied. \square

Notice that, whenever we have \mathcal{Q} containing 2-lame queries, like in Subsection IV-C, some sterile 2-lame red s-piders will appear in $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$.

OUTLINE

The queries in \mathbb{F}_s (or functions, depending on what level of abstraction one wants to see them) which we considered so far were unary, in the sense that they acted on single s-piders. In the rest of the paper we want them to be binary, so that they can **rewrite words from \mathbb{A}_s^* in a context-sensitive way**. And the ability to encode such a rewriting is a key to undecidability.

V. BINARY QUERIES

We will now define two operations – $\hat{\wedge}$ and $\hat{\vee}$ – each of them taking two queries from \mathbb{F}_s and returning new “binary” query, from the set that we will call \mathbb{F}_s^2 .

It is maybe good to recall here what are the free variables of the s-pider queries from \mathbb{F}_s : 2-lame queries have two free variables, and 1-lame queries have one: the free variables are the knees of the legs with missing calves. When a s-pider query f is seen as a green-red TGD, the free variables are what connects the new part of the structure, added by a single execution of a TGD, to the the old part⁷.

Definition 18: For $f, f' \in \mathbb{F}_s$ consider disjoint copies \mathbb{G} of $A[f]$ and \mathbb{G}' of $A[f']$. Let V and V' be the sets of elements

⁷They of course also connect via the constants.

of \mathbb{G} and \mathbb{G}' and let W and W' be the subsets of V and V' consisting of free variables of f and f' . Let z_2 and z_1 be the antenna and tail of f and let z'_2 and z'_1 be the antenna and tail of f' . Let $U(f, f')$ be the (disjoint) union of \mathbb{G} and \mathbb{G}' . Then:

- $f \wedge f'$ is the unique conjunctive query whose canonical structure is $U(f, f')$, with z_2 and z'_2 identified, and with the set of free variables equal to $W \cup W' \cup \{z_1, z'_1\}$;
- $f \vee f'$ is the unique conjunctive query whose canonical structure is $U(f, f')$, with z_1 and z'_1 identified, and with the set of free variables equal to $W \cup W' \cup \{z_2, z'_2\}$.

The set of all $f \wedge f'$ (or $f \vee f'$) for $f, f' \in \mathbb{F}_s$ will be called \mathbb{F}_s^\wedge (resp. \mathbb{F}_s^\vee). We also define \mathbb{F}_s^2 as $\mathbb{F}_s^\wedge \cup \mathbb{F}_s^\vee$.

The main lemma, obviously implying Theorem 4 is:

Lemma 19: *It is an undecidable problem to determine, for given $s \in \mathbb{N}$ and given $Q \subseteq \mathbb{F}_s^2$ whether $\text{Chase}(\mathcal{T}_Q, \star) \models \star$.*

The rest of the paper is devoted to the proof of this lemma.

VI. ABSTRACTING FROM THE S-PIDER DETAILS

Let $Q \subseteq \mathbb{F}_s^2$ be a set of binary queries. We would like to understand the structure of $\text{Chase}(\mathcal{T}_Q, \star)$ so that, in particular, we understand when $\text{Chase}(\mathcal{T}_Q, \star) \models \star$ holds.

First of all notice that Lemma 13 and 14 survive in the new context – together with their proofs. But be careful here:

Lemma 20: *For each pair a_1, a_2 of elements of $\text{Chase}(\mathcal{T}_Q, \star)$ there are at most two elements a such that $\text{Chase}(\mathcal{T}_Q, \star) \models H(a, a_1, a_2)$.*

Proof: Induction. For the induction step notice that an atom $H(a, a_1, a_2)$ can only be created together with either a new element a_1 (if a TGD generated by a query from \mathbb{F}_s^\vee is used) or with a new a_2 (when the query is from \mathbb{F}_s^\wedge). And notice that a single execution of a TGD generated by a query from \mathbb{F}_s^2 creates two atoms of the predicate H , and the newly created a_1 occurs in both of them. Notice that the newly created spiders are always both of the same color. \square

A. Queries $f_1 \wedge f_2$ and $f_1 \vee f_2$ in action

Let now $Q \in \mathcal{Q}$ be of the form⁸ $f_1 \wedge f_2$ and suppose \mathbb{D} is a structure (a stage of $\text{Chase}(\mathcal{T}_Q, \star)$).

Consider the TGD $Q^{R \rightarrow G}$ and let us try to imagine how this TGD could be executed in \mathbb{D} . First a homomorphism h from $A[R(Q)]$ to \mathbb{D} needs to be found.

$A[R(Q)]$ contains 3 antenna/tail vertices: z_1, z'_1 and z_2 , joined by the atoms $R(H)(z, z_1, z_2)$ and $R(H)(z', z'_1, z_2)$. This means that two red atoms $R(H)(h(z), h(z_1), h(z_2))$ and $R(H)(h(z'), h(z'_1), h(z_2))$ must be found in \mathbb{D} .

Notice that, due to Lemma 20, once $h(z_1)$, $h(z'_1)$ and $h(z_2)$ are fixed there are at most two possible choices for each of $h(z)$ and $h(z')$. And once $h(z)$ and $h(z')$ are fixed then, due to Lemma 14 there is exactly one real s-pider \mathcal{S}_1

in \mathbb{D} with $R(H)(h(z), h(z_1), h(z_2))$ and exactly one real s-pider \mathcal{S}_2 in \mathbb{D} with $R(H)(h(z'), h(z'_1), h(z_2))$.

Now, in order for the query $A[R(Q)]$ to be executed we need the query $R(f_1)$ to match with \mathcal{S}_1 and the query $R(f_2)$ to match with \mathcal{S}_2 . Lemma 11 tells us when it is possible.

Once the triples $h(z), h(z_1), h(z_2)$ and $h(z'), h(z'_1), h(z_2)$, satisfying all the above constraints, are found, a copy of $A[G(Q)]$ is created⁹, consisting of two green¹⁰ s-piders $f_1(\mathcal{S}_1)$ and $f_2(\mathcal{S}_2)$. This is because on the level of individual s-piders we are exactly in the world of Section IV.

What is new is how the two new s-piders are connected to each other and to the old part of the structure: the antenna of $f_1(\mathcal{S}_1)$ is a new element – it was quantified in Q – and is identified with the antenna of $f_2(\mathcal{S}_2)$.

But the tail of $f_1(\mathcal{S}_1)$ was free in Q so it is identified with $h(z_1)$ and the tail of $f_2(\mathcal{S}_2)$ was free and it is identified with $h(z'_1)$. So the new copy of $A[G(Q)]$ is connected to the old structure via the tails of the two new s-piders.

Of course the two new s-piders are also connected to the old structure via the free variables (and constants) which are not in their H atoms. But there are two reasons why we do not need to bother about it. First of them is Lemma 14. Second is that, while each TGD generated by a query from \mathbb{F}_s^2 needs two spiders to be executed, and requires them to share their antennas (or tails), it is oblivious to any other possible connections between the two s-piders (via knees). This analysis shows that we now can completely abstract from the low-level implementation details of the s-piders, in particular from details like the relations C^i , C_i , T^i , T_i and concentrate on the high-level notions.

B. S-warm and s-warm rewriting rules

A s-warm is defined as a multi-labeled graph (which means that each edge can have one or more labels), whose edges are the H atoms of some structure (intended to be a stage of $\text{Chase}(\mathcal{T}_Q, \star)$):

Definition 21: *A s-warm \mathbb{D} is a ternary relation $H \subseteq \mathbb{A}_s \times D \times D$. To keep notations light we use the term "elements of \mathbb{D} " for elements of D . Elements of \mathbb{A}_s are labels. We assume that for each two elements a, b of a s-warm there are at most two s-piders \mathcal{S} such that $H(\mathcal{S}, a, b)$, and that they are of the same color. Atoms $H(\mathcal{S}, a, b)$, or just pairs a, b , such that $\mathbb{D} \models H(\mathcal{S}, a, b)$, for some \mathcal{S} , are called edges. An edge is green or red, depending on the s-piders being its labels.*

We are going to see queries from \mathbb{F}_s^2 as s-warm rewriting rules:

Definition 22: *Let $Q = f \wedge f'$ (or $Q = f \vee f'$) be from \mathbb{F}_s^2 and let \mathbb{D} be a s-warm. We say that a rewriting Q can be executed in \mathbb{D} if:*

⁹Unless it already existed.

¹⁰Of course this is all true also for the colors reversed.

⁸The case when Q is of the form $f_1 \vee f_2$ is analogous.

③ there are edges $H(\mathcal{S}, a, b)$ and $H(\mathcal{S}', a', b)$ (resp. $H(\mathcal{S}, a, b)$ and $H(\mathcal{S}', a, b')$), such that $\mathcal{S} \in \text{Dom}(f)$ and $\mathcal{S}' \in \text{Dom}(f')$ are both of the same color;

④ there is no b' (resp. a') such that $H(f(\mathcal{S}), a, b')$ and $H(f'(\mathcal{S}'), a', b')$ (resp. $H(f(\mathcal{S}), a', b)$ and $H(f'(\mathcal{S}'), a', b')$) are edges of \mathbb{D} .

Pair of edges $H(\mathcal{S}, a, b)$ and $H(\mathcal{S}', a', b)$ is called the input of the rewriting (notice that order is important here). The result of the rewriting is then a new structure \mathbb{D}' being \mathbb{D} with new vertex b' (resp. a') and new edges $H(f(\mathcal{S}), a, b')$ and $H(f'(\mathcal{S}'), a', b')$ (resp. $H(f(\mathcal{S}), a', b)$ and $H(f'(\mathcal{S}'), a', b')$) as above.

Notice that we did not require in the above definition that $a \neq a'$ (resp. $b \neq b'$). Not only we have no means to enforce such requirement, but also, since we begin the Chase from a single (full green) s-pider the possibility of having them equal is of crucial importance for us.

See that – while we are not literally talking about TGDs now – conditions ③ and ④ are analogous to ① and ② from Section II-B and we can still (like in Section II-C) define a fair (with respect to a set \mathcal{Q} of rewritings and an original s-warm \mathbb{D}) sequence of structures $\{\mathcal{C}_k\}_{k \in \Omega}$, with each \mathcal{C}_k being a result of a single execution of a rewriting rule in the structure $\bigcup_{l < k} \mathcal{C}_l$ and with each possible rewriting ultimately being executed. We can also define the fixpoint of the rewritings, as the union of $\bigcup_{k \in \Omega} \mathcal{C}_k$. To distinguish, we will call the union *chase*(\mathcal{Q}, \mathbb{D}).

C. The abstraction

Let \mathbb{D} be a structure over Σ , such that if $\mathbb{D} \models H(a, b, c)$ then there is exactly one real s-pider \mathcal{S} in \mathbb{D} having a as its head, and such that each real s-pider in \mathbb{D} is isomorphic to some element of \mathbb{A} . The following definition and lemma hardly come as a surprise:

Definition 23: The s-warm s-warm(\mathbb{D}) is defined as the set of all triples $H(\mathcal{S}, b, c)$ such that $\mathbb{D} \models H(a, b, c)$ and a is the head of a real s-pider in \mathbb{D} which is isomorphic to \mathcal{S} .

From now on let \mathbb{D}_{\star} be the s-warm consisting of a single edge labeled with \star . Define \mathcal{F} as the set of all fair (with respect to the set $\mathcal{T}_{\mathcal{Q}}$ of TGDs and the structure \star) sequences $\{\mathcal{D}_k\}_{k \in \Omega}$ and let \mathcal{F}' be the set of all fair (with respect to the set \mathcal{Q} of rewriting rules and the s-warm \mathbb{D}_{\star}) sequences $\{\mathcal{C}_k\}_{k \in \Omega}$.

Lemma 24: The mapping that maps a sequence $\{\mathcal{D}_k\}_{k \in \Omega}$ of structures to a sequence $\{s\text{-warm}(\mathcal{D}_k)\}_{k \in \Omega}$ of s-warms is a bijection from \mathcal{F} to \mathcal{F}' .

For a given set $Q \subseteq \mathbb{F}_s^2$ let $\mathcal{C}^Q = \text{chase}(\mathcal{Q}, \mathbb{D}_{\star})$. From now on we forget about $\text{Chase}(\mathcal{T}_{\mathcal{Q}}, \star)$ and TGDs and concentrate on s-warms and their rewritings. Due to Lemma 24, in order to prove Lemma 19 it is enough to show:

Lemma 25: It is an undecidable problem to determine, for given $s \in \mathbb{N}$ and given set $Q \subseteq \mathbb{F}_s^2$ of rewriting rules,

whether \mathcal{C}^Q contains any edge labeled with \star .

D. One more lemma

Before we end this Section it will be maybe illuminating to notice one peculiar property of \mathcal{C}^Q . The proof of the following lemma goes by easy induction:

Lemma 26: Let $Q \subseteq \mathbb{F}_s^2$. Then each vertex of \mathcal{C}^Q either has in-degree zero (such vertex will be called tail, as it is the tail of all the edges it belongs to) or has out-degree zero (and it is the antenna of all the edges it belongs to). This implies that all the directed H -paths in \mathcal{C}^Q have length one.

Now imagine vertices of \mathcal{C}^Q drawn in two rows – all the antennas in the upper row and all the tails in the lower one – and see how mnemonic the fonts \uparrow and ∇ are.

VII. AN IMPORTANT EXAMPLE (QUITE COMPLICATED)

Consider a set \mathcal{Q}_{η} consisting of the following three pairs of associated rewritings:

- ① **A:** $\mathbb{f}_1 \uparrow \mathbb{f}_2$ and **B:** $\mathbb{f}_1^{\alpha} \uparrow \mathbb{f}_2^{\eta_1}$
- ② **A:** $\mathbb{f}_3^{\eta_0} \uparrow \mathbb{f}_4$ and **B:** $\mathbb{f}_3^{\beta_0} \uparrow \mathbb{f}_4^{\eta_1}$
- ③ **A:** $\mathbb{f}_5^{\eta_1} \nabla \mathbb{f}_6$ and **B:** $\mathbb{f}_5^{\beta_1} \nabla \mathbb{f}_6^{\eta_0}$

where $\alpha, \beta_0, \eta_0, \beta_1, \eta_1 \in \mathbb{S}$,

And let us try to have a glimpse of $\mathcal{C}^{\mathcal{Q}_{\eta}}$. Let $H(\star, s_0, t_0)$ be the only edge of \mathbb{D}_{\star} .

The table describes a (finite prefix of) an infinite sequence of rewritings that will be of special importance for us. Newly created elements are marked with bold.

Input edges	Rule used	Output edges
$H(\star, s_0, t_0), H(\star, s_0, t_0)$	①A	$H(\star_1, s_0, t'), H(\star_2, s_0, t')$
$H(\star_1, s_0, t'), H(\star_2, s_0, t')$	①B	$H(\star^{\alpha}, s_0, t_1), H(\star^{\eta_1}, s_0, t_1)$
$H(\star^{\eta_1}, s_0, t_1), H(\star, s_0, t_0)$	②A	$H(\star_5, s', t_1), H(\star_6, s', t_0)$
$H(\star_5, s', t_1), H(\star_6, s', t_0)$	②B	$H(\star^{\beta_1}, s_1, t_1), H(\star^{\eta_0}, s_1, t_0)$
$H(\star^{\eta_0}, s_1, t_0), H(\star, s_0, t_0)$	③A	$H(\star_3, s_1, t''), H(\star_4, s_0, t'')$
$H(\star_3, s_1, t''), H(\star_4, s_0, t'')$	③B	$H(\star^{\beta_0}, s_1, t_2), H(\star^{\eta_1}, s_0, t_2)$
$H(\star^{\eta_1}, s_0, t_2), H(\star, s_0, t_0)$	②A	$H(\star_5, s'', t_2), H(\star_6, s'', t_0)$

Compare the two rewritings using the rule ②A and notice the recursion. Then proving the following lemma will be an easy exercise:

Lemma 27: There are infinite sequences t_1, t_2, \dots and s_1, s_2, \dots of elements of $\mathcal{C}^{\mathcal{Q}_{\eta}}$ such that: $H(\star^{\alpha}, s_0, t_1)$, and for each k there is $H(\star^{\beta_1}, s_k, t_k)$ and $H(\star^{\beta_0}, s_k, t_{k+1})$ and $H(\star^{\eta_1}, s_0, t_k)$.

VIII. FRIENDLY THUE SYSTEMS

We will now consider Thue systems $\Pi \subseteq \mathbb{S}^* \times \mathbb{S}^*$. Elements of \mathbb{S} are numbers, so some of them are even and some are odd. We think that α, β_0 and η_0 are even and β_1 and η_1 are odd.

A set of productions of a Thue system $\Pi \subseteq \mathbb{S}^* \times \mathbb{S}^*$ will be called *friendly* if $\Pi = \Pi_{<} \cup \Pi_{=}$ where:

- $\Pi_{<}$ consists of two pairs $\{\eta_0, \beta_0 \eta_1\}$ and $\{\eta_1, \beta_1 \eta_0\}$;

- all productions of $\Pi_{=}$ are of the form $\{ij, i'j'\}$ for $i, i', j, j' \in \mathbb{S}$;
- if $\{ij, i'j'\} \in \Pi_{=}$ then both i and i' are odd and both j and j' are even or both i and i' are even and both j and j' are odd;
- there is no production of $\Pi_{=}$ of the form $\{ij, ij'\}$ or $\{ij, i'j\}$;
- there is no production involving α ; no production in $\Pi_{=}$ involves η_0 or η_1 ;
- there is an odd $\gamma \in \mathbb{S}$, and even $\gamma' \in \mathbb{S}$ which occur (each of them) in exactly one production of Π , which is $\{ii', \gamma\gamma'\}$ for some $i, i' \in \mathbb{S}$;
- $s > 2|\Pi|$

It is easy to prove (using the techniques presented in [Dav77]) that the problem:

For a set of productions of a friendly Thue system Π , do there exist $\mathbf{w}, \mathbf{w}' \in \mathbb{S}^*$ such that $\mathbf{w}\gamma\gamma'\mathbf{w}' \xrightarrow{*}_{\Pi} \alpha\eta_1$?

is undecidable.

Let now Π be a fixed friendly Thue system. Lemma 25, and therefore Theorem 4, will be proved once we construct a set \mathcal{Q} of rewritings such that the two conditions are equivalent:

- ⊙: There is an edge, in $\mathcal{C}^{\mathcal{Q}}$ labeled with \star ;
- ⊙: $\mathbf{w}\gamma\gamma'\mathbf{w}' \xrightarrow{*}_{\Pi} \alpha\eta_1$ for some $\mathbf{w}, \mathbf{w}' \in \mathbb{S}^*$.

The following Lemma is easy to prove and will be useful:

Lemma 28: Condition ⊙ holds if and only if there is $m \in \mathbb{N}$ such that $\mathbf{w}\gamma\gamma'\mathbf{w}' \xrightarrow{*}_{\Pi} \alpha(\beta_1\beta_0)^m\eta_1$ for some $\mathbf{w}, \mathbf{w}' \in \mathbb{S}^*$.

A. The set \mathcal{Q}

First we define \mathcal{Q}_0 , as the set of rewritings consisting of:

- all the rewriting rules from \mathcal{Q}_η
- two associated rewriting rules ④ $\mathbb{f}_{l_p}^i \wedge \mathbb{f}_{r_p}^j$ and $\mathbb{f}_{l_p}^{i'} \wedge \mathbb{f}_{r_p}^{j'}$ for each production $p = \{ij, i'j'\}$ in Π with i even;
- two associated rewriting rules ⑤ $\mathbb{f}_{l_p}^i \vee \mathbb{f}_{r_p}^j$ and $\mathbb{f}_{l_p}^{i'} \vee \mathbb{f}_{r_p}^{j'}$ for each production $p = \{ij, i'j'\}$ in Π with i odd;

where each of the numbers l_p and each r_p only occurs in the two aforementioned rewriting rules. Finally, \mathcal{Q} is defined as \mathcal{Q}_0 with one additional rewriting rule ⑥ $\mathbb{f}^\gamma \vee \mathbb{f}^{\gamma'}$ with r not occurring anywhere in the rules of \mathcal{Q}_0 .

OUTLINE

The rest of the paper is devoted to understanding the structure, first of $\mathcal{C}^{\mathcal{Q}_0} = \text{chase}(\mathcal{Q}_0, \mathbb{D}_\star)$ and then of $\mathcal{C}^{\mathcal{Q}} = \text{chase}(\mathcal{Q}, \mathbb{D}_\star)$ in order to prove that ⊙ holds true if and only of ⊙ does. In the following Section IX we prove that condition ⊙ implies ⊙.

IX. HOW TO HUNT A (FULL) RED S-PIDER

Definition 29: For a s -warm \mathbb{D} the set $\bar{W}(\mathbb{D}) \subseteq \mathbb{S}^* \times \mathbb{D}^2$ is defined as the smallest set such that:

- $\langle \varepsilon, a, a \rangle \in \bar{W}(\mathbb{D})$ for each $a \in \mathbb{D}$;
- if $\langle w, a, b \rangle \in \bar{W}(\mathbb{D})$ and $\mathbb{D} \models H(\star^i, b, b')$ for some even i then $\langle wi, a, b' \rangle \in \bar{W}(\mathbb{D})$;
- if $\langle w, a, b \rangle \in \bar{W}(\mathbb{D})$ and $\mathbb{D} \models H(\star^i, b', b)$ for some odd i then $\langle wi, a, b' \rangle \in \bar{W}(\mathbb{D})$.

Then $W(\mathbb{D}) \subseteq \mathbb{S}^*$ is defined as $\{w : \exists a, b \langle w, a, b \rangle \in \bar{W}(\mathbb{D})\}$.

In other words $W(\mathbb{D})$ is the set of all words that can be constructed as follows: walk an undirected path in \mathbb{D} (from some a to some b) and read (and remember) the labels of all the edges you cross. But you are only allowed to take edges labeled by green 1-lame s-piders. And if the label is \star^i , for an even i , then you must walk in the direction of the edge, otherwise you must walk in the opposite direction.

Lemma 30: If $\mathbf{v} \in W(\mathcal{C}^{\mathcal{Q}_0})$ and $\mathbf{v} \xrightarrow{*}_{\Pi} \mathbf{v}'$ then also $\mathbf{v}' \in W(\mathcal{C}^{\mathcal{Q}_0})$.

Proof: By induction it is enough to prove that if $\mathbf{v} \in W(\mathcal{C}^{\mathcal{Q}_0})$ and $\mathbf{v} \xrightarrow{*}_{\Pi} \mathbf{v}'$ then also $\mathbf{v}' \in W(\mathcal{C}^{\mathcal{Q}_0})$. Suppose that $\mathbf{v} = \mathbf{w}_1 i j \mathbf{w}_2$, that $\mathbf{v}' = \mathbf{w}_1 i' j' \mathbf{w}_2$ with $\{ij, i'j'\} \in \Pi_{=}$ and i even (the other case is analogous) and that $\mathbf{v} \in W(\mathcal{C}^{\mathcal{Q}_0})$. The last assumption means that there are vertices a, b, c, d, e in $\mathcal{C}^{\mathcal{Q}_0}$ such that both the triples $\langle \mathbf{w}_1, a, b \rangle$ and $\langle \mathbf{w}_2, d, e \rangle$ are in $\bar{W}(\mathcal{C}^{\mathcal{Q}_0})$ and edges $H(\star^i, b, c)$ and $H(\star^j, d, c)$ are in $\mathcal{C}^{\mathcal{Q}_0}$.

By the assumption that $\{ij, i'j'\} \in \Pi_{=}$ and i even we have that the rewritings $\mathbb{f}_{l_p}^i \wedge \mathbb{f}_{r_p}^j$ and $\mathbb{f}_{l_p}^{i'} \wedge \mathbb{f}_{r_p}^{j'}$ are in \mathcal{Q} . But – since $\mathcal{C}^{\mathcal{Q}_0}$ is closed under rewritings – the first of these rewritings enforces that there must be a c' in $\mathcal{C}^{\mathcal{Q}_0}$ with $H(\star_{l_p}, b, c')$ and $H(\star_{r_p}, d, c')$. And the second of the rewritings enforces that there must be a c'' in $\mathcal{C}^{\mathcal{Q}_0}$ with $H(\star^{i'}, b, c'')$ and $H(\star^{j'}, d, c'')$. So also $\mathbf{v}' \in W(\mathcal{C}^{\mathcal{Q}_0})$. \square

Lemma 31: Condition ⊙ implies ⊙.

Proof: By Lemma 28 condition ⊙ implies that there is $m \in \mathbb{N}$ such that $\mathbf{w}\gamma\gamma'\mathbf{w}' \xrightarrow{*}_{\Pi} \alpha(\beta_1\beta_0)^m\eta_1$ for some $\mathbf{w}, \mathbf{w}' \in \mathbb{S}^*$.

It follows from Lemma 27, and from Definition 29 that $\alpha(\beta_1\beta_0)^m\eta_1 \in W(\mathcal{C}^{\mathcal{Q}_0})$ for some $m \in \mathbb{N}$. By Lemma 30 this implies that the word $\gamma\gamma'$ is in $W(\mathcal{C}^{\mathcal{Q}_0})$, which means that there are vertices a, a', b of $\mathcal{C}^{\mathcal{Q}_0}$ such that $H(\star^\gamma, a, b)$ and $H(\star^{\gamma'}, a, b')$ hold in $\mathcal{C}^{\mathcal{Q}_0}$. Now use the rule $\mathbb{f}^\gamma \vee \mathbb{f}^{\gamma'}$ to produce an edge labeled with \star . \square

OUTLINE

Now we only need to prove that condition ⊙ implies ⊙. It is much more complicated than the opposite implication.

In the rest of the paper we assume that ⊙ does not hold true. Our goal is to show that ⊙ is not true either. The plan is to first consider a sequence $\{\mathcal{C}_i\}_{i \in \omega}$ (where ω is the

first infinite ordinal) fair (with respect to \mathcal{Q}_0 and \mathbb{D}_\star) and analyze the structure $\mathcal{C}^{\mathcal{Q}_0} = \bigcup_{i \in \omega} \mathcal{C}_i$. This will be done in Sections X–XII.

Then we must of course face the possibility that the list $\text{ToDo}(\textcircled{6}, \mathcal{C}^{\mathcal{Q}_0})$ will be very much non-empty and many (infinitely many) further rewritings may be needed. But – as we are going to prove in Section XIII – all the edges created by these rewritings will be sterile.

X. GETTING RID OF THE REDS

First of all notice that all the rewritings used in \mathcal{Q}_0 are lower¹¹. The proof of the following lemma is by (easy) induction, almost the same as the proof of Lemma 16:

Lemma 32: Let \mathcal{S} be the label of some edge in $\mathcal{C}^{\mathcal{Q}_0}$. Then \mathcal{S} is red if and only if it is lower. In particular \star is not a label of any edge in $\mathcal{C}^{\mathcal{Q}_0}$. Also \star_r cannot be a label of any edge in $\mathcal{C}^{\mathcal{Q}_0}$ (where r is from rule $\textcircled{6}$).

Definition 33: Two red edges $H(\mathcal{S}, a, b)$ and $H(\mathcal{S}', a', b)$ (or $H(\mathcal{S}, a, b)$ and $H(\mathcal{S}', a, b')$) of $\mathcal{C}^{\mathcal{Q}_0}$ will be called a married couple if they were created in the same rewriting step. The vertex b (resp. a) is called a knot then.

As it turns out, a knot is never touched by any edge other than the two spouses it joins:

Lemma 34: If a is a knot then it has degree 2 in $\mathcal{C}^{\mathcal{Q}_0}$.

Proof: Suppose the knot b was created, together with red edges $H(\mathcal{S}, a, b)$ and $H(\mathcal{S}', a', b)$ by an execution of some rule $\mathbb{F}_i^I \wedge \mathbb{F}_j^J$ (the case with $\mathbb{F}_i^I \vee \mathbb{F}_j^J$ is analogous). This rule was applied to two green edges with labels $\star^{I'}$, $\star^{J'}$ and thus $\mathcal{S} = \star_i^{I \setminus I'}$ and $\mathcal{S}' = \star_j^{J \setminus J'}$.

The only way to create a new edge containing the vertex b , would be to find some edge $H(\mathcal{S}_1, a, b')$ (or $H(\mathcal{S}_1, a', b')$) and use a rule of the form $f \vee f'$. But if f occurs in any rule from \mathbb{F}_s^Y then it cannot be applied to any spider of the form \star_i^K – this is because of the assumption that each of the lower subscripts can only occur in two associated rewritings. \square

Lemma 35 (No children out of wedlock, whatever temptation): Suppose $H(\mathcal{S}, a, b)$ is an element of a married couple of reds in some \mathcal{C}_n , created by an execution of some rule f from \mathcal{Q}_0 . Then the only way for it to be a part of the input of any future rewriting by a rule g from \mathcal{Q}_0 is that the other element of the input of this rewriting is its spouse and that g is the rule associated with f .

Proof: Suppose $f \in \mathbb{F}_s^Y$ (the other case is analogous), so a is the knot joining $H(\mathcal{S}, a, b)$ with its spouse. It follows from Lemma 34 that $g \in \mathbb{F}_s^Y$ – otherwise the degree of a would be greater than 2 at some point. So the only way for an edge to be an input of a rewriting together with $H(\mathcal{S}, a, b)$

is to contain the vertex a . But a only belongs to two edges in $\mathcal{C}^{\mathcal{Q}_0}$: to $H(\mathcal{S}, a, b)$ and its spouse. Using the argument from the proof of Lemma 34, that the numbers l_p and r_p can only occur in two associated rewritings, we get that g is either f itself (which is impossible due to idempotence) or is associated with f . \square

Notice (and this remark will be needed in Section XIII) that the proof does not rely on the shortage of possible candidates who would be keen to produce offspring with $H(\mathcal{S}, a, b)$. The reasons for its faithfulness is inherent to $H(\mathcal{S}, a, b)$ itself, and even someone like $H(\star, c, b)$ would not change its mind (\star being the most promiscuous red label).

Lemma 36 (Sterile reds): (i) If a red $H(\mathcal{S}, a, b)$ is an element of a married couple of red edges in some \mathcal{C}_n and \mathcal{S} is 2-lame then neither $H(\mathcal{S}, a, b)$ nor its spouse are never used as an input of a rewriting rule execution.

(ii) A rewriting in which \star is used as an input of any 2-lame rewriting rule from \mathbb{F}_s leads to a pair of sterile red edges.

Proof of this Lemma is left as an easy exercise. Use the assumption that there is no production in Π of the form $\{ij, ij'\}$ or $\{ij, i'j'\}$ and the argument from the proof of Lemma 17.

From now on we assume that the sequence $\{\mathcal{C}_k\}_{k \in \mathbb{N}}$ is such, that whenever a married couple of reds is created at some step, at the next step the only rewriting this red marriage is able to be the input of is executed (unless this married couple is sterile). So we can imagine that we always execute procedures consisting of two associated rules, and produce greens from other greens. The red edges are in the structure but in no way they contribute to its complexity and we do not need to think of them any more.

XI. DANGEROUS VERTICES

In our quest to understand the structure of $\mathcal{C}^{\mathcal{Q}_0}$ we now concentrate on the green edges. We already know (from Lemma 36) that 2-lame rewritings applied to \star never produce anything relevant. Notice also that all the rewritings used in \mathcal{Q}_0 are lower, and all green edges of $\mathcal{C}^{\mathcal{Q}_0}$, apart from edges labeled with \star , are upper. This means that only 2-lame rewriting rules can be applied to edges with labels of the form \star^i .

In particular this means that any rewriting with the rule $\textcircled{1}A$ must take as its input two edges labeled with \star , rewritings $\textcircled{2}A$ and $\textcircled{3}A$ take one edge labeled with \star and one with \star^i and so on.

Definition 37: A vertex of \mathcal{C} , or any \mathcal{C}_i , is called dangerous if it is a tail or an antenna of some edge labeled with \star .

Lemma 38: Let $H(\star^i, a, b)$ be a green edge of \mathcal{C}_i . Then (i) a is dangerous if and only if i is either α or η_0 and (ii) b is dangerous if and only if i is η_1 .

Proof: By induction on i . The claim is clearly true in \mathcal{C}_0 as it consists of a single edge labeled with \star .

¹¹It is not true about \mathcal{Q} and this is the reason why we analyze $\mathcal{C}^{\mathcal{Q}_0}$ first.

Suppose the claim is true in some \mathcal{C}_n . Suppose the structure \mathcal{C}_{n+2} is a result of first applying some rewriting f to green edges $H(\mathcal{S}_1, a, b)$ and $H(\mathcal{S}_2, a', b)$, (or to $H(\mathcal{S}_1, a, b)$ and $H(\mathcal{S}_2, a, b')$ – in cases where rewriting rules ③ or ⑤ were used), creating two new red edges, and then applying f' , associated with f , to the two new red edges (we know, from Section X, that this is the only scenario one needs to consider).

As a result a new vertex b' (resp. a') is created, together with new green edges $H(\mathcal{S}'_1, a, b')$ and $H(\mathcal{S}'_2, a', b')$, (resp. $H(\mathcal{S}'_1, a', b)$ and $H(\mathcal{S}'_2, a', b')$). We need to check that a and a' (resp. b and b') do not become dangerous in \mathcal{C}_{n+2} (if they were not in \mathcal{C}_n) and that the new edges and new vertex do not contradict the claim. There are now, unfortunately, 8 cases we need to inspect, depending on f and f' :

- f, f' of the form ④. Then each of $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}'_1, \mathcal{S}'_2$ is of the form \star^i for some $i \neq \alpha, \eta_0, \eta_1$. By assumption none of a, a', b is dangerous in \mathcal{C}_n and they remain non-dangerous in \mathcal{C}_{n+2} . The new b' is non-dangerous either. The claim holds in \mathcal{C}_{n+2} .

- f, f' of the form ⑤. Analogous to the previous case.

- f, f' of the form ①A, ①B. Then $\mathcal{S}_1 = \mathcal{S}_2 = \star$ so a, a' and b are all dangerous. $\mathcal{S}'_1 = \star^\alpha$ and $\mathcal{S}'_2 = \star^{\eta_1}$ and the new b' is non-dangerous in \mathcal{C}_{n+2} . The claim holds in \mathcal{C}_{n+2} .

- f, f' of the form ①B, ①A. Then $\mathcal{S}_1 = \star^\alpha$ and $\mathcal{S}_2 = \star^{\eta_1}$ so, by assumption, a and a' must have already been dangerous in \mathcal{C}_n . $\mathcal{S}'_1 = \mathcal{S}'_2 = \star$ and so the new b' is created as dangerous. The claim holds in \mathcal{C}_{n+2} .

- f, f' of the form ②A, ②B. Then $\mathcal{S}_1 = \star^{\eta_0}$ and $\mathcal{S}_2 = \star$. By assumption b and a' are dangerous but a is not. $\mathcal{S}'_1 = \star^{\beta_0}$ and $\mathcal{S}_2 = \star^{\eta_1}$, so b' is non-dangerous and a remains non-dangerous in \mathcal{C}_{n+2} . The claim holds in \mathcal{C}_{n+2} .

- f, f' of the form ②B, ②A. $\mathcal{S}'_1 = \star^{\beta_0}$ and $\mathcal{S}_2 = \star^{\eta_1}$. By assumption a' is dangerous while a and b are not. $\mathcal{S}'_1 = \star^{\eta_0}$ and $\mathcal{S}_2 = \star$ so b' is created as dangerous but a remains non-dangerous in \mathcal{C}_{n+2} . The claim holds in \mathcal{C}_{n+2} .

The two cases with ③ are analogous to the cases with ②. \square

XII. CHARACTERIZATION OF $W(\mathcal{C}^{\mathcal{Q}_0})$

A word $i_0 i_1 \dots i_{l-1} i_l \in \mathbb{S}^*$ is *correct* if for all $k \neq 0$ there is $i_k \neq \alpha$ and for all $k \neq l$ there is $i_k \neq \eta_0$ and $i_k \neq \eta_1$. A word $i_0 i_1 \dots i_{l-1} i_l \in \mathbb{S}^*$ is *maximal correct* if it is correct and $i_0 = \alpha$ and $i_l = \eta_0$ or $i_l = \eta_1$.

Lemma 39: (i) For each correct $\mathbf{w} \in W(\mathcal{C}^{\mathcal{Q}_0})$ there is a maximal correct $\mathbf{v} \in W(\mathcal{C}^{\mathcal{Q}_0})$ such that \mathbf{w} is a subword of \mathbf{v} .

(ii) If $\mathbf{v} \in W(\mathcal{C}^{\mathcal{Q}_0})$ is maximal correct then $\mathbf{v} \stackrel{}{\Leftrightarrow}_{\Pi} \alpha \eta_1$.*

Proof: It is enough to prove that both claims hold in each $W(\mathcal{C}_n)$, and this can be proved by induction. The claim is clearly true in \mathcal{C}_0 as $W(\mathcal{C}_0)$ is empty. The induction step follows the proof of Lemma 38, and similar case inspection is needed. For f, f' of the form ④ apply the argument from the proof of Lemma 30.

Also for both the ② and both the ③ cases the validity of the induction hypothesis for $n+2$ follows from the assumption about $W(\mathcal{C}_n)$ and the fact that the word $\mathbf{w} \in W(\mathcal{C}_{n+2})$ under consideration is a result of one rewriting, using one of the rules from $\Pi_{<}$, and applied to some word in \mathcal{C}_n .

In the case of f, f' of the form ①B, ①A no new words are added to $W(\mathcal{C}_{n+2})$. Finally, in the case of f, f' of the form ①A, ①B one new correct word is created¹². It is $\alpha \eta_1$, which clearly satisfies both the claims of the Lemma. \square

Now it easily follows from Lemma 39 that:

Lemma 40: No edge in $\mathcal{C}^{\mathcal{Q}_0}$ is labeled with \star^γ or with $\star^{\gamma'}$

XIII. FROM $\mathcal{C}^{\mathcal{Q}_0}$ TO $\mathcal{C}^{\mathcal{Q}}$

For each pair of edges¹³ of the form $H(\star, a, b), H(\star, a, b')$ in $\mathcal{C}^{\mathcal{Q}_0}$ let us now apply a rewriting using the rule ⑥. Each such rewriting will result in adding a new vertex a' and new edges $H(\star^\gamma, a', b)$ and $H(\star^{\gamma'}, a', b')$. Call the resulting structure \mathcal{C} . Notice that all the new vertices of \mathcal{C} are of degree 2.

Proof of Lemma 31, and thus of Lemma 25 and Theorem 4, will be completed once we show:

Lemma 41: (i) $\text{ToDo}(Q, \mathcal{C})$ is empty. In consequence $\mathcal{C} = \mathcal{C}^{\mathcal{Q}}$. (ii) There is no edge labeled with \star in \mathcal{C} .

Proof: Claim (ii) is obvious – there was no such edge in $\mathcal{C}^{\mathcal{Q}_0}$ and we never added one while building \mathcal{C} on the top of $\mathcal{C}^{\mathcal{Q}_0}$. For the proof of Claim (i) first notice that no rewriting with green inputs is possible in \mathcal{C} :

- no such rewriting using rules from \mathcal{Q}_0 is possible since \mathcal{C} has no new green edges compared to $\mathcal{C}^{\mathcal{Q}_0}$ and

- no such rewriting using rule ⑥ and at least one 1-lame green edge is possible, since no edge of $\mathcal{C}^{\mathcal{Q}_0}$ is labeled with \star^γ or $\star^{\gamma'}$ (Lemma 40), and

- no such rewriting using rule ⑥ and both inputs labeled with \star is possible any more – by the definition of \mathcal{C} .

Now how about the possibility of rewritings in \mathcal{C} using red edges as the input? No such rewriting using rules of \mathcal{Q}_0 and having, as the input, at least one red edge from $\mathcal{C}^{\mathcal{Q}_0}$ is possible, by Lemma 35 (however tempting the new red edges would look!). By Lemma 32 neither \mathbb{f}^γ nor $\mathbb{f}_r^{\gamma'}$ match with any red edge from $\mathcal{C}^{\mathcal{Q}_0}$. This means that no red edge from $\mathcal{C}^{\mathcal{Q}_0}$ can be an input of any new rewriting in \mathcal{C} .

To finish the proof notice that none of the rewritings from \mathbb{F}_s^λ can use either \star^γ or $\star^{\gamma'}$ as one of its inputs. Since all the new edges in \mathcal{C} are of degree 2 the only rule from \mathbb{F}_s^λ that matches with the new edges of \mathcal{C} is ⑥, which however cannot be used due to idempotence reasons. \square

¹²What is actually created is a new copy of this word.

¹³In formal terms this means that we extend the fair sequence $\{\mathcal{C}_n\}_{n \in \omega}$ with new structures. We do not rely on that so we do not need to prove it, but there are infinitely many of the new structures, as there are infinitely many edges in $\mathcal{C}^{\mathcal{Q}_0}$ labeled with \star . Thus the new fair sequence is $\{\mathcal{C}_n\}_{n \in 2\omega}$. The structure \mathcal{C} is now – as always – defined as $\bigcup_{n \in 2\omega} \mathcal{C}_n$.

REFERENCES

- [AD98] Serge Abiteboul and Oliver M. Duschka. Complexity of answering queries using materialized views. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 254–263, New York, NY, USA, 1998. ACM.
- [Afr11] Foto N. Afrati. Determinacy and query rewriting for conjunctive queries and views. *Theor. Comput. Sci.*, 412(11):1005–1021, March 2011.
- [Dav77] M. Davis. Unsolvability problems. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 567–594. North-Holland, Amsterdam, 1977.
- [DPT99] Alin Deutsch, Lucian Popa, and Val Tannen. Physical data independence, constraints, and optimization with universal plans. In Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik, and Michael L. Brodie, editors, *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, pages 459–470. Morgan Kaufmann, 1999.
- [FG12] Enrico Franconi and Paolo Guagliardo. The view update problem revisited. *CoRR*, abs/1211.3016, 2012.
- [FGZ12] Wenfei Fan, Floris Geerts, and Lixiao Zheng. View determinacy for preserving selected information in data transformations. *Inf. Syst.*, 37(1):1–12, March 2012.
- [FKN13] Enrico Franconi, Volha Kerhet, and Nhung Ngo. Exact query reformulation over databases with first-order and description logics ontologies. *J. Artif. Intell. Res. (JAIR)*, 48:885–922, 2013.
- [Hal01] Alon Y. Halevy. Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294, December 2001.
- [JK82] D. S. Johnson and A. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. In *Proceedings of the 1st ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, PODS '82, pages 164–169, New York, NY, USA, 1982. ACM.
- [LMS95] Alon Y. Levy, Alberto O. Mendelzon, and Yehoshua Sagiv. Answering queries using views (extended abstract). In *Proceedings of the Fourteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '95, pages 95–104, New York, NY, USA, 1995. ACM.
- [LY85] Per-Ake Larson and H. Z. Yang. Computing queries from derived relations. In *Proceedings of the 11th International Conference on Very Large Data Bases - Volume 11*, VLDB '85, pages 259–269. VLDB Endowment, 1985.
- [NSV07] Alan Nash, Luc Segoufin, and Victor Vianu. Determinacy and rewriting of conjunctive queries using views: A progress report. In Thomas Schwentick and Dan Suciu, editors, *Database Theory – ICDT 2007*, volume 4353 of *Lecture Notes in Computer Science*, pages 59–73. Springer Berlin Heidelberg, 2007.
- [NSV10] Alan Nash, Luc Segoufin, and Victor Vianu. Views and queries: Determinacy and rewriting. *ACM Trans. Database Syst.*, 35:21:1–21:41, July 2010.
- [Pas11] Daniel Pasailă. Conjunctive queries determinacy and rewriting. In Tova Milo, editor, *Proceedings of the 14th International Conference on Database Theory (ICDT'11)*, pages 220–231, Uppsala, Sweden, March 2011. ACM Press.
- [SV05] Luc Segoufin and Victor Vianu. Views and queries: Determinacy and rewriting. In *Proceedings of the Twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '05, pages 49–60, New York, NY, USA, 2005. ACM.
- [YL87] H. Z. Yang and Per-Ake Larson. Query transformation for psj-queries. In *Proceedings of the 13th International Conference on Very Large Data Bases*, VLDB '87, pages 245–254, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.